Diseño y Evaluación de Configuraciones

Traffic Control:
Control de tráfico en Linux

J.M. Drake

Notas:

Objetivo del control de tráfico

- El control de tráfico es el mecanismo de que se dispone en las redes basadas en paquetes para regular el tráfico de información en la red y con ellos garantizar que los diferentes flujos disponen de los recursos que necesitan para garantizar sus calidades de servicio: Latencia, Throughput, fiabilidad, y costo.
- Existen dos estrategias:
  - Servicios integrados (*intServ*): Es un mecanismo de gestión fina de la QoS basada en que cada flujo reserva en cada Router de la red los recursos necesarios para que la comunicación se realice con las características requeridas.
  - Servicios diferenciados (*difServ*): es un mecanismo de gestión grosera de la QoS basada en limitar los flujos que acceden a la red, y con ello garantizar la disponibilidad de recursos que requiere cada flujo.
- El control de tráfico es el mecanismo que tiene Linux para implementar la estrategia de servicios diferenciados.

Notas:

Objetivos concretos de Traffic Control

- Limitar la anchura de banda que puede utilizar un nudo, una aplicación o una conexión.
- Reserva una fracción de la anchura de banda para un nudo una aplicación o una conexion.
- Maximizar el flujo de tráficos en una red asimétrica, por priorización de unos paquetes respectos de otros.
- Garantizar latencia en tráfico sensible o urgente.
- Gestión eficiente de la anchura de banda sobrante.
- Eliminar ciertos tipos de tráficos no permitidos.

- Conviene recordar que el control del tráfico no resuelve todos los problemas. A veces lo que se necesita es mayor anchura de banda.

Notas:

Ventajas y desventajas del control de tráfico

- Ventajas:
  - De forma muy simple y eficiente permite que el flujo de información se haga predecible, y la disponibilidad de los recursos sea gestionables.
  - Se ordena el tráfico de forma que flujos masivos dispongan de la BW que necesitan, flujos urgentes tengan la latencia que necesitan y flujos sin requisitos específicos dispongan de los recursos que se necesitan.
  - Somete a los clientes a disciplinas predefinidas, que los clientes no tienen que conocer.
- Desventajas:
  - Es un sistema globalizado y muy complejo. Cuando el sistema es grande se necesitan aplicar reglas genéricas.
  - Si se usa inadecuadamente conduce a una reducción drástica de recursos.
  - Supone una sobrecarga sobre el sistema de comunicaciones, que en sistemas complejos puede ser significativos.
  - El control que lleva a acabo es siempre grosero, y si se necesita un control mas fino puede ser que sea necesario un sistema integrado.

Notas:

Flujos y colas.

- Un flujo es una conexión o sesión de comunicación entre dos aplicaciones. En TCP y UDP sobre IP, se identifica por las dos parejas (*Dirección Ip* y *Puerto)* de cada uno de los procesadores que se comunican.
  - Los flujos son los que se encuentran caracterizados en un sistema, y llevan asociados requisitos de QoS de acuerdo con su naturaleza.
- Las colas son los mecanismos básicos que se utilizan en la planificación a los recursos limitados.
  - Son los elementos en los que los paquetes se reordenan, reagrupan, retrasan descartan o priorizan.
  - Desde el punto de vista de las aplicaciones sólo existe una cola de transmisión. Se cuando se diseña el detalle del servicio de comunicaciones cuando se organizan múltiples colas y se cualifican sus características y políticas.

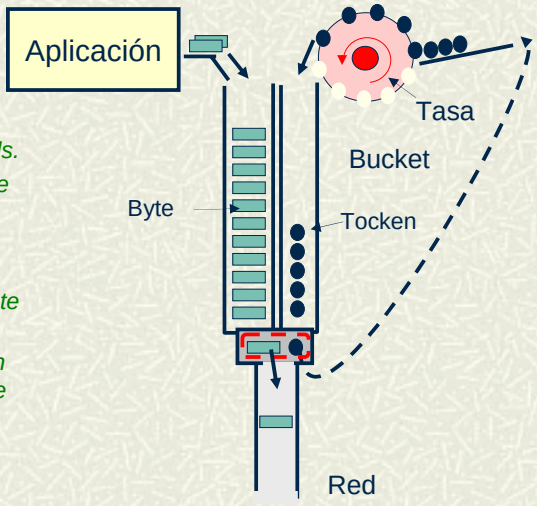
Notas:

Tockens y Buckets

- El modelo Tockens Buckets Filter (TBF) es el mecanismo simple y eficiente que se utiliza para limitar de forma flexible el flujo de información:

El algoritmo puede ser explicado como sigue:

- *Un tocken es añadido al bucket cada 1 / rate seconds.*
- *El bucket sólo puede contener un número limitado de tockens. Si un tocken llega cuando el bucket está lleno, se descarta.*
- *Cuando un paquete con n bytes llega de una aplicación, se retiran n tocken del bucket, y el paquete se envía por la red.*
- *Si hay menis de n tokend en el bucket, so se extraen tocken y el envío de paquete se retrasa hasta que se hayan acumulados suficientes tokens en el bucket..*



Notas:

Packets y frames

- En los sistemas de comunicación hay tres unidades de información:
  - El paquete (*packet*) es una unidad de información del nivel del transporte (capa 3ª de la pila OSI). Su longitud es variable, aunque puede estar limitada.
  - El marco (*frame*) es una unidad de información del nivel de enlace /capa 2º de la pila OSI). Su longitud es definidad por la interfaz que se utiliza. En el caso de Ethernet Hay cabecera, y datosútiles entre 0 y 1500 bytes.
  - El byte es un octeto de bits con capacidad de almacenar un código entre 0 y 255.

Notas:

Operaciones básicas del control de tráfico (1)

- Ajuste (*Shaping*): Es el mecanismo por el que un paquete es retrasado en la cola de salida transmisión para ajustar el flujo de salida a la tasa establecida.
  - Es la operación básica para limitar la anchura de banda que utiliza el flujo que se transmite.
  - Es una operación *no conservativa* y por tanto requiere de un buffer (*queue*) para soportarlo.
  - Conlleva el incremento de la latencia del flujo sobre el que opera, a cambio sirve para garantizar la latencia del resto de los flujos.
- Planificación (*Scheduling*): Reorganiza los paquetes de la cola de acuerdo con una cierta política de planificación:
  - FIFO: First Input first output.
  - SFQ (Stochastic Fair Queieing): Planifica los paquetes de forma que se transmitan equitativamente entre todos los flujos
  - WRR (Wide Round-Robin): Asigna a los paquetes de cada flujo tiempos igualesde transmisión.
  - GREF(Generic Random Early Drop):
  - ESFQ: (Extended Stochastic Fair Queieing): Permite controlar el algoritmo de hashing.

Notas:

Operaciones básicas del control de tráfico (2)

- Clasificador (*Classifying*): El clasificador ordena o clasifica el flujo en diferentes flujos que son gestionados con diferentes características.
  - La clasificación se puede realizar por diferentes criterios: por las marcas que llevan los paquetes, por la dirección de destino, por la dirección de origen, etc.
  - En control de tráfico, la clasificación se raliza mediante cascadas de filtros.
- Supervisión (*Policing*): En esta operación se ejecutan medidas y en función de ellas se limita el tráfico en las colas.
  - Es un mecanismo acepta tráfico a una cierta tasa, y el tráfico que sobrepasa el valor especificado o bien se reclasifica, o bien se elimina.
  - Se utiliza para garantizar que los flujos entre interlocutores se mantienen a las tasas que admiten el mas lento.
  - Son operaciones de alternativas, sin capacidad de retrasar los paquetes.
- Eliminación (*Dropping*): Es la operación de eliminar, paquetes, flujos y clasificación
- Marcado (Marking):Es la operación a través de la que se modifica el contendo de un paquete.
  - TC modifica el campo DSCP (*DiffServ Code Point*), el cual es utilizado por otros elementos para gestionar el flujo en el proio TC , o en otros elementos (Router) de la red.

Notas:

Relaciones entre Elementos de TC y Operaciones de control

Operación	Componente de Linux TC
Shaping	***Class*** ofrece las capacidades de clasificación.
Scheduling	Un ***qdisc*** es un planificador. Los planificadores pueden ser simples como FIFO o complejos conteniendo clases y otros qdisc .
Classifying	Los ***filters*** realizan la clasificación a través de un agente interno.
Policing	Existe un ***policier*** en TC Linux que se implemnta como parte de un **filter**
Dropping	Se realiza a través de un *filter* con un agente *policer* que ejecuta la acción *drop*.
Marking	El ***dsmark qdisc*** puede utilizarse para marcar paquetes.

Notas:

Software y herramientas

- Las versiones de Linux posteriores a 2.2 tienen integrado en su núcleo las herramientas **iproute2** que integran **tc** (Traffic-Control).
- La herramienta **tc contiene todas las configuraciones** del núcleo requeridas para el control de tráfico.
- **tc** se puede gestionar desde la **línea de comandos** LARTC con una sintaxis anticuada que refleja la evolución histórica de la herramienta.

Ejemplo sencillo que usa tc para añadir una discilina de cola a la cola de un dispositivo

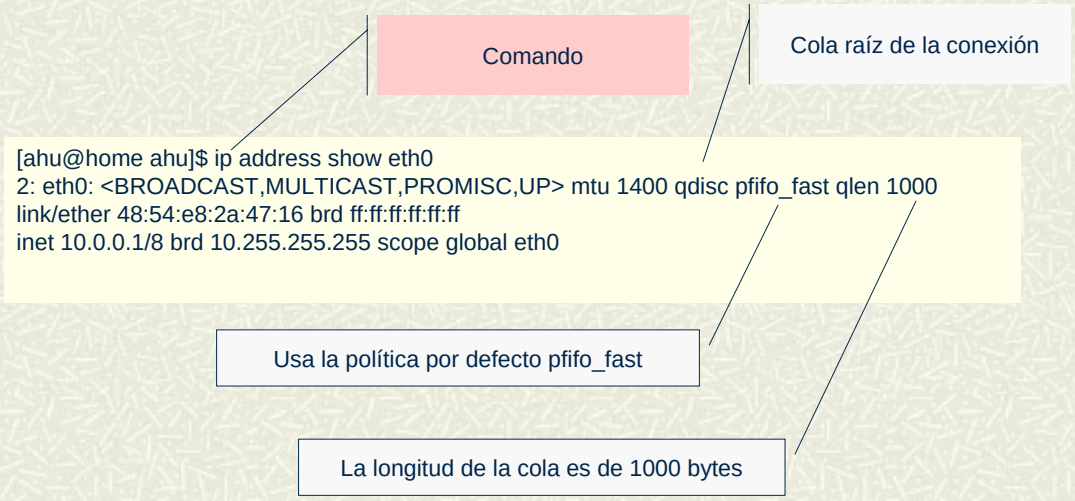
```
[root@luser1]# tc qdisc add \        Add a queuing discipline. The verb could also be del.
               > dev eth0 \          Specify the device onto which we are attaching the new
                                     queuing discipline

               > root \              The word rootmeans “egress” to tc.
               > handle 1:0 \        The handle is a user-specified number of the form major:minor.
               > htb                 This is the queuing discipline to attach, HTB in this example.
```

- Actualmente existe una nueva interfaz **tcng** (Traffic Control Next Generation) mas moderna y que permite el control de trafico desde el código de programas.

Notas:

Cola por defecto de la conexión Ethernet

- Con el comando ip se puede obtener información de la conexión ethernet:



Notas:

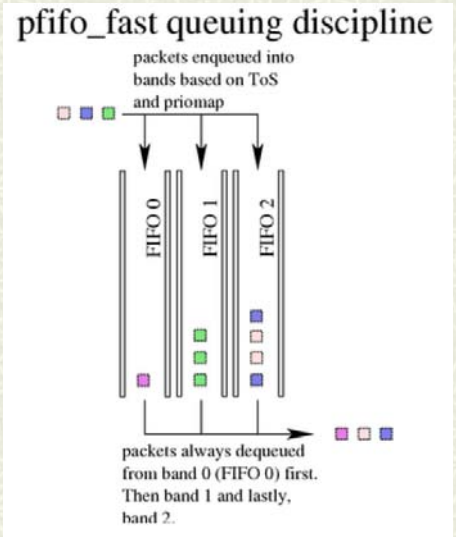
Disciplina de cola pfifo_fast

- Es una cola que usa la disciplina FIFO pero que utiliza tres bandas de prioridades (0 mas prioritaria, 1 y 2 menos prioritaria)
  - La distribución de los paquetes en las tres colas se realiza de acuerdo con el campo TOS del paquete.
  - No tiene parámetros de configuración.

Minimizar throughput
Minimizar retraso

TOS															
	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Banda	1	2	1	1	2	2	2	2	0	0	0	0	1	1	1	1

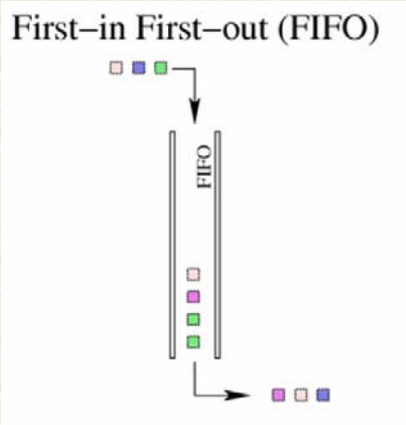
Minimizar costo
Minimizar fiabilidad



Notas:

Cola FIFO

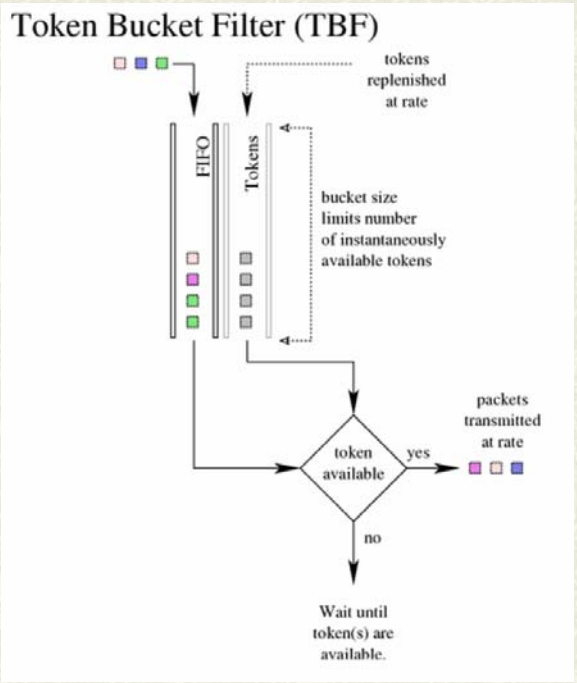
- Transmite los mensajes en el orden FIFO, esto es, en el orden exacto en que se han recibido.
  - Es la mas simple, pero lleva estadística de la longitud del buffer. Es un buen medio de conocer el nivel de carga de la cola.



Notas:

Disciplina TBF

- Permite limitar el proceso de desencolado de la información utilizando el algoritmo Tocken Bucket Filter
- Parámetros configuración:
  - **limit**: Nº de byte que pueden ser encolados a la espera de tocken.
  - **burst**: Yamaño del bucket.
  - **mpu**:Tamaño mínimo del paquete
  - **rate**:Tasa de generación de tocken.
  - **peakrate**: Velocidad a la que se envía los byte cuando hay tockens.
  - **mtu**:tamaño máximo de un frame.

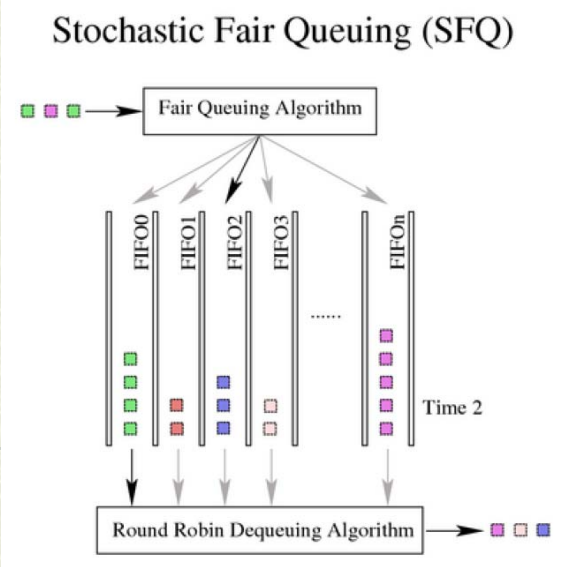


Notas:

Stochastic Fairness Queueing (SFQ)

- Utiliza múltiples colas a fin de distribuir la capacidad de transmisión entre todas las conexiones que transmiten.
  - Utiliza una tabla hash para distribuir conexiones con colas.

Parámetros de configuración:
- **perturb**: Tiempo entre reasignaciones de la tabla hash.
- **quantum**: Cantidad de byte que se sacan de una cola antes de pasar a la siguiente.
- **limit**: Total de bytes que peden ser encolados



Notas:

Ejemplos de asignación de una disciplina de cola:

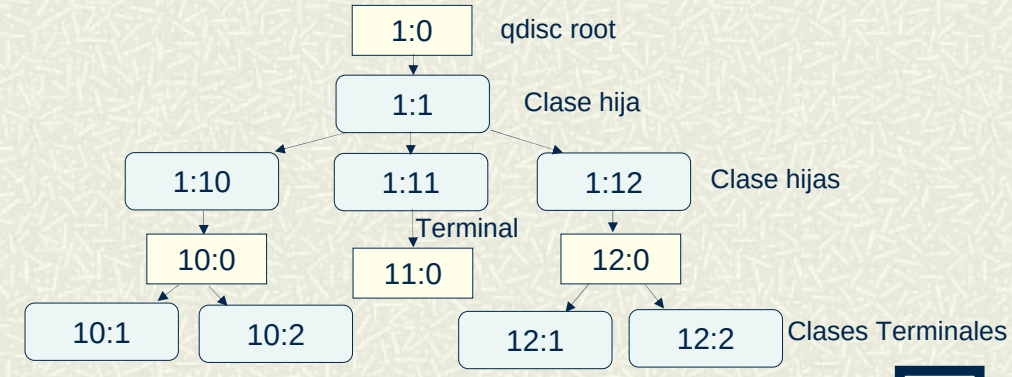
```
# tc qdisc add dev ppp0 eth0 tbf rate 220kbit latency 50ms burst 1540
#tc –s –d qdisc ls
qdisc tbf 8001:eth0 root rate 220000bit burst 1539b/8 mpu 0b lat 50ms
 Sent 4812 bytes 62 pkts (dropped 0, overlimits 0)
```

```
# tc qdisc add dev eth0 root sfq perturb 10
# tc -s -d qdisc ls
qdisc sfq 800c: dev eth0 quantum 1514b limit 128p flows 128/1024
perturb 10sec
Sent 4812 bytes 62 pkts (dropped 0, overlimits 0)
```


Notas:

Qdisc con clases

- A una qdisc se le pueden definir clases. Cada una de ellas representa una división interna a la que se pueden redirigir paquetes.
- Una clase a su vez puede contener otras clases . A la clase que no tiene definidas clases internas se denomina clase terminál.
- Las clases terminales tienen definida una qdisc interna



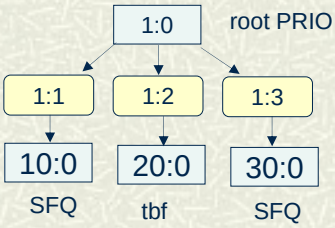
Notas:

Qdisc PRIO

- La qdisc PRIO no realiza ajustes, cada paquete que recibe lo reenvía a una calse hija de acuerdo con la configuración de sus filtros.
- Cada qdisc PRIO tiene varias clase internas la x:1, la x:2 .... Cuando recibe una orden de desencolar, primero pregunta a la x:0, y si esta tiene paquete es el que se desencola. En caso contrario pregunta a las otras por orden.
- Las clases internas tienen por defecto una qdisc FIFO. Pero su disciplina se puede cambiar.
- Los filtros pueden basarse en diferentes características de los paquetes, y no sólo de TOS.
- Parámetros de una qdisc PRIO:
  - Bands: Numero de bandas a crear
  - Priomap: Si no hay filtros se define un mapa de prioridades para interpretar el TOS.

Notas:

Ejemplo de configuración de una qdisc PRIO




```
# tc qdisc add dev eth0 root handle 1: prio
## Esto crea *instantáneamente las clases 1:1, 1:2, 1:3
# tc qdisc add dev eth0 parent 1:1 handle 10: sfq
# tc qdisc add dev eth0 parent 1:2 handle 20: tbf rate 20kbit buffer 1600 limit 3000
# tc qdisc add dev eth0 parent 1:3 handle 30: sfq
```

```
# tc -s qdisc ls dev eth0
qdisc sfq 30: quantum 1514b
Sent 0 bytes 0 pkts (dropped 0, overlimits 0)
qdisc tbf 20: rate 20Kbit burst 1599b lat 667.6ms
Sent 0 bytes 0 pkts (dropped 0, overlimits 0)
qdisc sfq 10: quantum 1514b
Sent 132 bytes 2 pkts (dropped 0, overlimits 0)
qdisc prio 1: bands 3 priomap 1 2 2 2 1 2 0 0 1 1 1 1 1 1 1 1
Sent 174 bytes 3 pkts (dropped 0, overlimits 0)
```


Notas:

Ejemplo de clasificación de paquetes con filtro.

```
# tc filter add dev eth0 protocol ip parent 10:  prio 1 u32 match ip dport 22 0xffff flowid 10:1
# tc filter add dev eth0 protocol ip parent 10:  prio 1 u32 match ip sport 80 0xffff flowid 10:1
# tc filter add dev eth0 protocol ip parent 10:  prio 2 flowid 10:2
```

- Coloca tres filtros en la qdisc 10:0 :
  - El flujo que se dirija hacia la puerta 22 se reenvía hacia la clase 10:1
  - El flujo que se dirija hacia la puerta 80 se reenvía hacia la clase 10:1
  - El resto del flujo hacia la clase 10:2

Notas:

Ordenes típicas de marcado

- La mayoría tiene una sintasis:

  # tc filter add dev eth0 parent 1:0 protocol ip prio 1 u32 ..

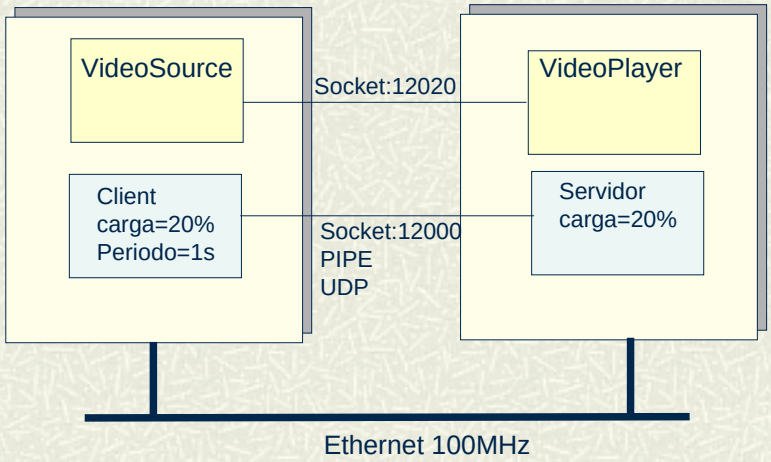
  Hacen referencia a u32 y pueden hacer referencia a cualquier prte de un paquete.

  - Sobre la dirección de origen/destino:
    - Máscara de origen «match ip src 1.2.3.0/24»
    - Máscara de destino «match ip dst 4.3.2.0/24»
  - Sobre puerto de origen/destino, todos los protocolos de IP
    - Origen: «match ip sport 80 0xffff»
    - destino: «match ip dport 0xffff»
  - Sobre protocolo IP (tcp, udp, icmp, gre, ipsec)
    - Use los números de /etc/protocols.

    Por ejemplo, icmp es 1: «match ip protocol 1 0xff».
- Existen otros muchos tipos de protocolos.

Notas:

Práctica



Plan de trabajo:

1º Ejecutar las dos aplicaciones

2º Incrementar la carga de la red aumentando la longitud del mensaje hasta que la QoS del vídeo se degrade.

3º Controlar el trafico de la red para conseguir que la QoS del vídeo no se afecte por la carga de la red.

Notas: